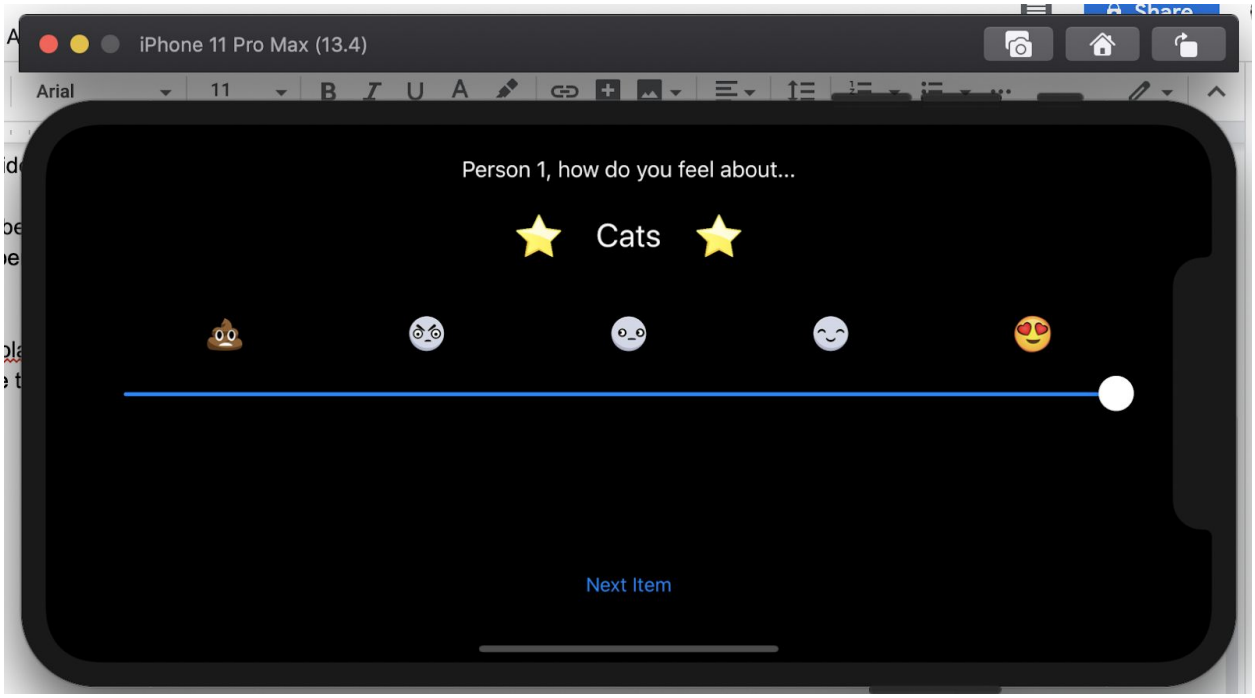
Project 1 - ComparisonShopper
Practice Debugging an existing app

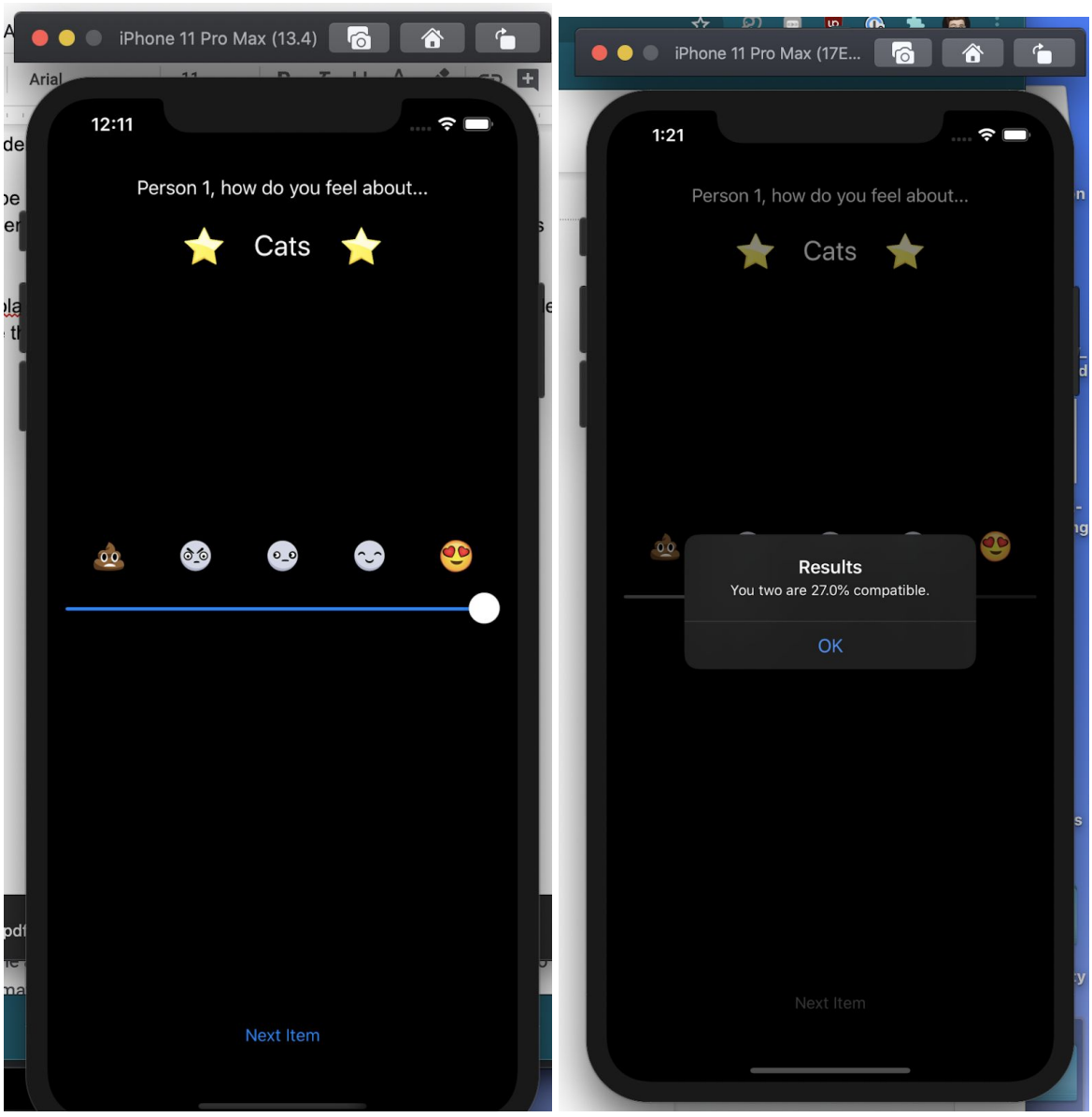
My app ComparisonShopper has 2 issues that need debugging.
1. It crashes when you open the app. Add breakpoints and use lldb to see if anything is nil. You can also use lldb commands after a crash to view values at the time of the crash.
2. If the user adds a second house to compare, none of the labels or images show up. Add breakpoints to make sure the correct code is running!

Project 2 - Compatibility Slider

Now, you're going to be building a mini dating app.
Make sure to open the Compatibility Slider starter project, I've created some classes and functions for you already.

1. Use your autolayout stack view and flexible view knowledge to line up the elements. Make the app look like this:





2. Person class
The idea of Compatibility Slider is that 2 users rate `compatibilityItems` and then the app will give them a percentage of how compatible they are.

Take a look at Person.swift. Read the notes about why Person is a class instead of a struct here.

Also, take a look at `items` in `Person`. Theoretically, if your only `compatibilityItems` are “Cats” and “Dogs” you could have a class that looks like:

```
class Person {
    var id: Int
```

```
    var catScore: Float
    Var dogScore: Foat
}
```

However, what if you wanted to add 10 more items to the array? You don't want to add 10 new properties to `Person,` that's extremely tedious. So instead, we're going to store a dictionary of ["itemNameFromArray": Float].

3. Time to make this app functional.
I wrote a function that will do the compatibility calculating for you, but it's up to you to implement the following logic.

- When the user opens the app, the top label should say "User 1, what do you think about..."
- The label with stars next to it should have the name of the first item in the `compatibilityItems` array.
- The user can move the slider and give that item a score between 1 and 5.
- When the user presses the "next item" button, you need to save their score for that first item as a dictionary.
    - Hint:

```
let currentItem = compatibilityItems[currentItemIndex]
currentPerson?.items.updateValue(slider.value, forKey: currentItem)
```

- After you save that info, go on to the next item in the array for Person1. A good way to keep track of what item you're on is to increase `currentItemIndex` after you save the info.
- When you get to the last item in the array, change the currentUser to person2, change the top label to say "Person 2, how do you feel about...", start the array over, and do the exact same thing as before.
- When you have finished with person 2, calculate both users compatibility scores and show a UIAlert telling the users their score.
- Reset the game to the beginning when the UIAlert comes up.
- Make sure to test your code a few times before you submit it, and watch out for crashes. If you try to access `compatibilityItems[currentItemIndex]` but `currentItemIndex` is a higher index than the last item in the array, you'll crash your app!